

NUMBER OF VARSIONS OF PROGRAM MANAGEMENT SYSTEM

Patent Number: JP61243533
Publication date: 1986-10-29
Inventor(s): ASAKAWA HIROHIKO
Applicant(s):: FUJI ELECTRIC CO LTD
Requested Patent: ☐ JP61243533
Application Number: JP19850083998 19850419
Priority Number(s):
IPC Classification: G06F9/06 ; G06F15/16
EC Classification:
Equivalents: JP1673558C, JP3037214B

Abstract

PURPOSE:To simplify software by collating the number of versions of data of a response message with the said number of data stored in a storage means by the first processor to permit cooperative operator of processor on condition that they coincide with each other.

CONSTITUTION:A master processor MP as the first processor and plural slave processors SP1-SPm as the second processors are connected to each other by a common bus CB of a system. A storage means M including a ROM and a RAM and a floppy disc device FD as the number of versions of storage means are connected to the bus CB, and the storage means M is connected to processors SP1-SPm. Version the number of data of the response message is collated with the number of versions of data stored in the storage means M by the processor MP, and cooperative operations of the processor MP and processors SP1-SPm are permitted on condition that they coincide with each other, thus simplifying software.

Data supplied from the esp@cenet database - I2

⑨ 日本国特許庁(JP)

⑩ 特許出願公開

⑫ 公開特許公報(A)

昭61-243533

⑬ Int.Cl.⁴

G 06 F 9/06
15/16

識別記号

庁内整理番号

7361-5B
2116-5B

⑭ 公開 昭和61年(1986)10月29日

審査請求 未請求 発明の数 1 (全9頁)

⑮ 発明の名称 プログラム版数管理方式

⑯ 特 願 昭60-83998

⑰ 出 願 昭60(1985)4月19日

⑱ 発 明 者 朝 川 裕 彦 川崎市川崎区田辺新田1番1号 富士電機株式会社内
⑲ 出 願 人 富士電機株式会社 川崎市川崎区田辺新田1番1号
⑳ 代 理 人 弁理士 山 口 巖

明 細 書

1. 発明の名称 プログラム版数管理方式

2. 特許請求の範囲

1) 共通のバスにより相互結合されそれぞれプログラムを格納する付属記憶手段をその固有のバス上に備えてデータ処理上相互に連携されたマルチプロセッサ系内のプログラムの版数を管理する方式であって、マルチプロセッサ系内の少なくとも1個の第1のプロセッサが、系全体に含まれる各プログラムの版数を記憶する不揮発性の版数記憶手段と、系内の他の第2のプロセッサに対してそのそれぞれが前記付属記憶手段内に格納するプログラムの版数を問い合わせるメッセージを発する照会メッセージ発生手段とを備え、該照会メッセージを受ける前記第2のプログラムが問い合わせに応じて該プロセッサの管理内にあるすべてのプログラムについて版数を示すデータを収集して前記第1のプロセッサに回答メッセージを発する回答メッセージ発生手段を備え、該回答メッセージ内の版数データを前記第1のプロセッサによりその

前記版数記憶手段に格納された版数データと照合し、両者が一致することを条件にマルチプロセッサ系内の連携動作を許可するようにしたことを特徴とするプログラム版数管理方式。

2) 特許請求の範囲第1項記載の方式において、第2のプロセッサが版数データを収集すべきプログラムに該第2のプロセッサに専ら付属するサブプロセッサの管理内にあるプログラムが含まれることを特徴とするプログラム版数管理方式。

3. 発明の詳細な説明

【発明の属する技術分野】

本発明は共通のバスにより相互結合されそれぞれプログラムを格納する付属記憶手段をその固有のバス上に備えてデータ処理上相互に連携されたマルチプロセッサ系内のプログラムの版数を管理する方式、とくに比較的複雑なマルチプロセッサ構成のPOS装置系のプログラム管理に有利な方式に関する。

【従来技術とその問題点】

上述のような共通バスで結ばれた例えばPOS

装置用のマルチプロセッサ系では、各プロセッサはそれぞれ付属の記憶手段としてROM、RAMあるいは補助記憶手段としてフロッピ装置を備え、プロセッサごとに割り当てられた所定のジョブ、例えばプリンタ、キーボード、POS端末機の制御等のジョブをその範囲内で主に果たすが、その記憶手段内のデータや場合によりプログラムは、マルチプロセッサ系内で転送ないしは流送されて系内の共有のデータないしは資産として使用に供される。マルチプロセッサ系の規模が大きくなり、プロセッサ内の一部が系の本体から離れた場所に置かれたり、その受け持つジョブが特殊化してきたり、新しいプロセッサが系に追加されたりすると、各プロセッサが実行するプログラムの統合性が系内において問題になってくる。

すなわち、専用化ないしは特殊化したジョブを受け持つプロセッサでは、常にその果たす機能の改良や新規な追加が必要であるから、そのつどプログラムの一部書き換えや全面書き直しが行われる。かかる、ソフトウェアの改良や追加はそれ自

- 3 -

持つ機能が多様化して来ると、系内で使用ないしは流送されるプログラムの数も多くなり、以上のような版数を揃えたりチェックしたりする作業自体が非常に厄介になって来る。系内の各プロセッサはしばしばそのサブプロセッサを従えており、このサブプロセッサが実行するプログラムに小規模な改良がなされた場合は、とくにこの版数の整合が見落され勝ちになる。従来この各プロセッサ間のプログラムの版数の整合はもっぱら人手に頼るのみであり、版数の整合の必要性が不注意により見落されたり、整合させる作業にもミスが生じたりし勝ちであり、ある一つのプロセッサに他のプロセッサと整合の取れない版数のプログラムがその結果としてロードされたにも拘らず、そのままマルチプロセッサ系が稼働状態に投入されてプログラムを暴走させた結果、系全体のシステムダウンが起こったり、重要な情報ファイル内のデータが破壊されたりする危険を避けることができなかった。

【発明の目的】

体は必要でもあり望ましいことなのであるが、そのソフトウェアによって作られたデータを系内の他のプロセッサによってデータ処理をする際に、それまでと同じやり方でデータ処理をすると結果が間違ったり誤差を生じたりすることになりやすい。プログラムを共用する場合は問題はさらに厄介になり、今までのプログラムとの整合性が失われると、プログラムを暴走させたりデータを破壊させたりしてしまうことになる。このため、系内のあるプログラムの改良や書き直しをした場合に、そのプログラムの版数いわゆるバージョン数を決めて改良や書き直しがなされたことがわかるようにするとともに、そのプログラムと整合を必要とするその他のプログラムも必要に応じて一部書き換えた上で新しい版数を決めてやるが行なわれる。もちろん、この他のプログラムの新しい版数は改良や書き直しが行なわれたプログラムの版数と合わせるのが最も簡単でもあり、管理上は最も確実である。

しかし、系の規模が大きくなりその各部が受け

- 4 -

本発明はマルチプロセッサ系内の各プロセッサがその付属記憶手段に格納しているプログラムの版数不一致による前述のような系の事故を予防して、ファイルの破壊を避ける上に有用なプログラム版数管理方式を得ることを目的とする。また、その副次的な目的は該管理方式をプログラム版数整合の作業のために役立てることにある。

【発明の要点】

本発明によれば上の目的は、系内の少なくとも1個のプロセッサ、例えばマスタプロセッサを第1のプロセッサとして、これに系全体に含まれる各プログラムの版数を記憶する不揮発性の版数記憶手段と、系内の他の第2のプロセッサに対してそのそれぞれが前記付属記憶手段内に格納するプログラムの版数を問い合わせるメッセージを発する照会メッセージ発生手段とを設け、この照会メッセージを受ける第2のプロセッサ、例えばスレーブプロセッサの例には、問い合わせに応じて該プロセッサの管理内にあるすべてのプログラムについて版数を示すデータを収集して前記第1のプ

ロセッサに回答メッセージを発する回答メッセージ発生手段を設けて、回答メッセージ内の版数データを前記第1のプロセッサによりその前記版数記憶手段に格納された版数データと照合し、両者が一致することを条件にマルチプロセッサ系内の連携動作を許可するように系を運営することによって達成される。

周知のようにマルチプロセッサ系内の各プロセッサは原理上は互いに平等な関係にあるが、各プロセッサの受け持つ役割りが分化するにつれて、その内の例えば1個をいわゆるマスタプロセッサとして、他をスレーブプロセッサとして運用することがよく行なわれる。この場合のスレーブプロセッサはそれぞれ専門の機能を受け持つ一方、マスタプロセッサはマルチプロセッサ系内の全体管理の機能を果たすことが多いから、前述の構成中の第1のプロセッサとしての機能はこのマスタプロセッサに持たせるのが合理的である。この第1のプロセッサが受け持つ版数記憶機能は電源が断のときにも失われてはならないから、版数データ

- 7 -

が保たれていてかつその各版数が正確に第1のプロセッサにより記憶されておれば足り、全体的な数字合わせやコード合わせの要は必ずしもなくなるから、プログラムの系内の整合性を保つための作業がそれだけ簡単になる。

本発明方式の実施は、系のいわゆるシステム立ち上げ時、例えばPOS装置系では朝の営業開始のための電源投入時に該システム立ち上げ動作の一環として行なうのがよい。回答メッセージ中の版数データと版数記憶手段内の版数データが一致しない場合、系の全体の起動を差し止めるなり、問題があるプロセッサ系の動作を部分的に差し止め、ないしはそれへのプログラムのロードを禁止するなり、あるいは少なくとも他のプロセッサとの連携動作を差し止めるなりすることにより、営業時間内に生じうる無用なトラブルを未然に防止できるからである。しかし、このほか本発明方式は、プログラムの修正、変更とそれに伴う版数の切換え時にも随時実施することにより、版数書き換えのエラー見落しを検出するためにも利用する

- 9 -

は不揮発性の記憶手段内に格納しておく必要があり、マスタプロセッサは系内の他の重要なデータや基本プログラムを管理しているから、これらと合わせて版数データを不揮発性記憶手段例えば付属のフロッピー装置や固定ディスク装置内に格納しておく機能を持たせるのが便利である。一方第2のプロセッサ、例えばスレーブプロセッサの方はそのサブシステムとしてサブスレーブプロセッサを持っていることがあり、プログラムの版数の系内の統合的な整合性を保つにはこのサブスレーブプロセッサの管理内にあるプログラムも管理の枠内に含めるべきであるから、第1のプロセッサからの照会メッセージに応じる第2のプロセッサからの回答メッセージにサブスレーブプロセッサ内のプログラムの版数データをも含ませるのが合理的である。

プログラムの版数を人手で管理する場合には、前述のようにすべてのプログラムの版数を統一的に同じとするのが最も間違いを少なくする上で望ましいが、本発明の場合はプログラム上の整合性

- 8 -

ことができ、これによってプログラムの修正、変更に伴う作業を確実にかつ容易にすることができる。POS装置の場合には、プログラムの変更や差し換え作業は営業時間外の限られた時間内に複数のプロセッサ側で同時に行なわれることがあり、相互の連絡にとかく手落ちが生じやすいから、本発明方式の実施は相互間の不一致を確実に検出できる点で実際の運営面から見ても非常に有用である。

【発明の実施例】

以下図を参照しながら本発明の実施例を詳しく説明する。第1図は本発明方式の全体構成図、第2図および第3図は該方式内でのメッセージの交流態様を示す図、第4図は該メッセージの構成態様を示す図である。

第1図において、本方式の第1のプロセッサとしての単一のマスタプロセッサMPと第2のプロセッサとしての複数個のスレーブプロセッサSP1(1~m)とは、並列な複数本からなる共通バスCBによって相互に連携され、一つのマルチプロセッ

サ系を構成している。また、スレーブプロセッサ中の若干、図では1番目のスレーブプロセッサSP1はその固有のバスIB、その入出力ポートIOおよび複数本のシリアルラインSLを介してサブスレーブプロセッサSSPJ($j=1\sim m$)と連携されている。共通バスCBのマスタプロセッサSPの側にはROMおよびRAMからなる記憶手段Mが設けられており、それらは元来スレーブプロセッサSP1にも共用のためのものであるが、本方式の実施上はその少なくとも一部はマスタプロセッサMPの専用エリアとして設定しておくのがよい。その右側に入出力ポートIOを介して接続された版数記憶手段10としてのフロッピディスク装置FDは、これもマルチプロセッサ系に共用でき、各プロセッサMPおよびSPによって実行されるプログラムなどが格納されているが、本発明方式の実施時、例えば系の立ち上げ時にあたっては、その中のプログラムのスレーブプロセッサSP1へロード後はマスタプロセッサMPに持たせるようにすべきである。もちろん、このフロッピディスク装置FDの中には、版数記憶

-11-

コード、IMbはメッセージの送り先のプロセッサを示す番号I、IMcは以降のメッセージ長を示すバイト数、IMdは該メッセージ内容照会のための問い合わせであることを示すコード、IMeはメッセージ内容の伝達の正確さを保つためのチェックコードである。この照会メッセージIMは、第2図に示すように本発明方式の動作スタートの直後に先頭のスレーブプロセッサSP1に共通バスCBを介して送られる。この照会メッセージIMを受ける側のスレーブプロセッサSP1も、その記憶手段MのROM内に回答メッセージを作るためのプログラムを固定的に記憶していて、これとともに回答メッセージ発生手段30を構成しており、第4図例に例示するような様式の回答メッセージAMを編纂、発生して、第2図に示すようにマスタプロセッサSPに返信する。この例における回答メッセージAMは8個の部分AMa~AMhを含み、これらの内AMaはメッセージ開始コード、AMbは回答元すなわち自己のプロセッサ番号I、AMcは以降のメッセージ長を示すバイト数、斜線を施されたAMdとAMeとはそれぞれそ

-13-

手段10として記憶エリアが設定されており、各プログラムの版数データは各プロセッサを示す前述の変数I、J($i=1\sim m, j=1\sim n$)ごとの表、例えば該2変数I、Jをアドレスとするいわば2次元アレイの形で記憶される。もちろん、各プロセッサが実行すべきプログラムは一般的には複数個あるから、この場合にはさらに独立の第3の変数kをアドレスとして版数をプログラムごとに記憶させればよい。以下、説明を簡単化するために、版数データはプロセッサごとにはそのROM内のプログラムの版数とRAM内に納めるべきプログラムの版数がそれぞれ1個ずつであるとする。

この実施例における照会メッセージ発生手段10は図示のようにマスタプロセッサMPとROM内のその専用エリアM1とからなり、該ROMエリアM1内に固定記憶されているプログラムに従って第4図例に例示するような照会メッセージIMを発生する。この例における照会メッセージIMはその先頭から末尾に至るまでのIMa~IMeで示された5個の部分からなり、内IMaはメッセージ開始を示す

-12-

のROMとRAM内のプログラムの版数VN、最後のAMhはチェックコードである。

この回答メッセージAMを受けたマスタプロセッサMPは、その中に含まれた送り元のプロセッサ番号Iに対応する版数のデータを版数記憶手段10から読み出し、その読み出しデータと回答メッセージAM内に含まれている版数VNとを照合して、両者が一致することを確認する。このプロセスが第2図でチェックプロセスCBとして示されている。この確認結果が正であれば、同様に次のスレーブプロセッサSP2への照会メッセージの送付、回答メッセージAMの受け取りおよびチェックプロセスCBが繰り返される。第2図ではスレーブプロセッサSP3からの回答メッセージAM中の版数VNが版数記憶手段10からの読み出しデータと不一致のため、システムの立ち上げがその段階でストップされたことが示されている。

前述のようにスレーブプロセッサSP1の若干はサブスレーブプロセッサSSPJ($j=1\sim n$)を従えており、この場合の回答メッセージAMを編纂する

-14-

プロセスが第3図に、回答メッセージAHの形式例が第4図向に示されている。この場合のスレーブプロセッサSP1は照会メッセージ発生手段10としても機能するようにその付属ROMにプログラムがなされており、第3図に示すようにサブスレーブプロセッサSSPJ($j=1 \sim n$)に順次第4図向に示した照会メッセージIHを渡し、同図向に示した回答メッセージAHを受取ってそれらのメッセージ内容から回答メッセージAHを編纂して最終的に回答メッセージAHをマスタプロセッサMPに送る。この際、スレーブプロセッサSP1とこれと距離を隔てたサブスレーブプロセッサSSPJ($j=1 \sim n$)とが前述のようにシリアルラインSLで結ばれている場合は、メッセージの送達と受理の正確さを期すためにそれらを確認する確認メッセージCHを交換するようにするのが良い。また、スレーブプロセッサSP1ではサブスレーブプロセッサSSPJから回答メッセージAHを受けても、その中の版数をチェックする作業は一切行わず、チェックプロセスCHはもっぱらマスタプロセッサMP側に一任する。

-15-

回答メッセージAH中の版数VNと版数記憶手段10内の対応する版数データとの一致を確認し、その結果がすべて良であれば始めて系の運転を許可する。この許可は第1図に示すように例えばそのスレーブプロセッサSP2の記憶手段MにプログラムPGをフロッピディスクの装置PDからロードするのを許可したり、スレーブプロセッサSP1の記憶手段M内のデータDを他のプロセッサに転送するのを許可したりする形で行なわれる。

最後に第5図に示すフローを参照しながら、本発明方式におけるプロセッサ側の動作を説明する。図の縦の一点鎖線の左側はマスタプロセッサMP側の動作を、右側はスレーブプロセッサSP側の動作をそれぞれ示す。

マスタプロセッサMPのスタートは、例えばマルチプロセッサ系のシステム立ち上げの一環であり系への電源投入を契機とする。最初のステップS1では、問い合わせ先として1番目のスレーブプロセッサSP1を宛換1に1を置くことによって指定する。次のステップS2ではこの1番目のスレーブ

-17-

スレーブプロセッサSPからマスタプロセッサMPに送る回答メッセージAHの内容については、第4図向に示すようにその最初の5個の部分は前と同じAHa~AHeの部分であり、これにサブスレーブプロセッサSSPJに関するn個の部分が続けられる。この各サブスレーブプロセッサSSPJに関する部分は、それぞれそのサブスレーブプロセッサの番号jを示す部分AHbjとそのROM、RAM中のプログラムの版数VNをそれぞれ示す部分AHdj, AHejの3個の部分からなる。これらのサブスレーブプロセッサSSPJ($j=1 \sim n$)に関する部分の次は、前と同様なチェックコード部分AHfによって回答メッセージAHを終える。マスタプロセッサMPでは、その送り元スレーブプロセッサ番号部AHb中の数字1と、サブスレーブプロセッサ番号部AHbj中の数字jとをアドレスとして、版数記憶手段10から版数データを読み出す。

以上のようにして、マスタプロセッサMPはスレーブプロセッサSP1($i=1 \sim m$)と逐次に照会メッセージIHおよび回答メッセージAHを交換した上、

-16-

プロセッサSP1に照会メッセージAHを発信する。ステップS3は回答メッセージAHを持ち、回答メッセージAHを受けるとステップS4でそのメッセージ長部に指定された回答メッセージAHが全部受信されたか否かをチェックする。この受信完了後のステップS5では、そのスレーブプロセッサSP1にサブスレーブプロセッサSSPJが付属しているかどうかを例えば回答メッセージAHの内容から判断して、否であれば次のステップS6に、正であれば左上方のステップS8に移行する。否である場合のステップS6では版数記憶手段10から宛換1に対応する版数データを読み出し、次のステップS7で回答メッセージ中の版数VNとそれが一致するかどうかを判断する。もし不一致であればフローは直ちに左下のステップS15に移って不一致であったスレーブプロセッサの番号1を表示するなどの警告を発した後にフローを中断する。

前の判断ステップS5でサブスレーブプロセッサSSPJがあることがわかると、ステップS8でサブプロセッサ番号jに1を置いて、続くステップS9、

-18-

S10において1番目のサブプロセッサから順次にサブスレーブプロセッサSSPjに対する版数VNと記憶版数データとを比べて、不一致であれば前と同様にステップS15でそのサブスレーブプロセッサ番号jをスレーブプロセッサ番号1とともに表示した上でフローを中断する。判定結果が一致であればステップS11でサブスレーブプロセッサの指定番号1を歩進させ、ステップS12で最後のサブスレーブプロセッサSSPnについて判定が完了したか否かを確かめ、否であればフローをステップS9に戻す。サブスレーブプロセッサ全部について判定が良であればフローはステップS12からステップS13に移り、次のスレーブプロセッサ番号に変換1を歩進させる。

このようにして全部のスレーブプロセッサSPI(1=1~m)およびそれらに付属するサブスレーブプロセッサSSPjについて版数VNがすべて正しいことがわかれば、S14から抜け出して全フローを終え、システムの動作開始を前述のような態様で許可する。なお、以上の動作フローの説明におい

-19-

確認メッセージCHを取る要があるので、ステップS18でメッセージを発したのちのステップS19はこの確認メッセージCHの待機ステップになり、所定の短時間後にもこの確認メッセージCHがサブスレーブプロセッサSSPjから来ない場合は、ステップS18に戻って照会メッセージIHを発信し直す。確認メッセージCHが取れたときには、前と同様にステップS20, S21においてサブスレーブプロセッサSSPjからの回答メッセージAHを待ち、かつそれが終了したことを確認する。続くステップS22では、前に第3図例で示した回答メッセージAH中のチェックコード部AHfにより該メッセージが変形されていなかったかどうかを確かめ、変形ありの場合は再度フローをステップS18に戻して回答メッセージAHの再発信を要請する。ステップS22でのチェック結果が良のときには、第3図に示したようにその確認メッセージCHをサブスレーブプロセッサSSPjに発信してやる方がよいのであるが、必ずしもその必要があるわけではないので第5図のフローには含まれていない。ステップS23では

-21-

て、ステップS6およびS7で版数データをそのつど版数記憶手段10から読み出すこととしたが、実際上は最初のステップS1の直前にすべてのプロセッサ番号1,jについてあらかじめ一括して版数データを読み込んでおき、RAM等に記憶させておくのが便利であることはもちろんである。

スレーブプロセッサSPI(1=1~m)における動作フローは、前述のステップS2で発信される照会メッセージIHを受けることによりスタートされる。その最初のステップS10からはサブスレーブプロセッサSSPの有無によってフローが分かれ、なしの時はステップS20に飛んで回答メッセージAHを発信するだけでフローを終えるが、ありの場合はS17~S25のサブスレーブプロセッサとのメッセージ交換に入る。ステップS17ではサブスレーブプロセッサ番号jに1をセットし、これによって指定されたサブスレーブプロセッサSSPjにステップS18で照会メッセージIHを発する。サブスレーブプロセッサとの交信は前に第3図について説明したようにメッセージが到達しない場合に備えて

-20-

回答メッセージAH中のそのサブスレーブプロセッサ番号jに対応する版数VNを記憶し、つづくステップS24で変換1をインクリメントさせ、かつステップS25で全部のサブスレーブプロセッサとの交信を終えたかどうかを判定して、然りであれば前述のステップS26において第4図例に示した形式の回答メッセージAHを編纂してマスタプロセッサMPに送る。

【発明の効果】

以上説明したように本発明によれば共通のバスにより相互結合されそれぞれプログラムを格納する付属記憶手段をその固有のバス上に備えてデータ処理上相互に連携されたマルチプロセッサ系内のプログラムの版数を管理する方式として、まず該マルチプロセッサ系内の少なくとも1個のプロセッサ、例えばマスタプロセッサを第1のプロセッサとして、これに系全体に含まれる各プログラムの版数を記憶する不揮発性の版数記憶手段と、系内の他の第2のプロセッサに対してそのそれぞれが前記付属記憶手段内に格納するプログラムの

-22-

版数を問い合わせるメッセージを発する照会メッセージ発生手段とを設け、該第1のプロセッサから照会メッセージを受ける側の第2のプロセッサ、例えばスレーブプロセッサには、問い合わせに応じて該プロセッサの管理内にあるすべてのプログラムについて版数を示すデータを収集して前記第1のプロセッサに回答メッセージを発する回答メッセージ発生手段を設け、回答メッセージ内の版数データを前記第1のプロセッサによりその前記版数記憶手段に格納された版数データと照合し、両者が一致することを条件にマルチプロセッサ系内の連携動作を許可するようにしたので、従来まったく人手に頼っていたプログラムの版数の整合確認作業をマルチプロセッサ系内に短時間内で自動的にすませることができるようになり、系内のプログラムの暴走やこれに基づくデータファイル等の破壊事故を未然にかつ確実に予防することができる。この本発明方式はプログラムの版数を整合させる作業そのものに対してもそのための有用なツールとして利用することができ、整合作業を

-23-

面で例示するフロー図である。図において、

10: 版数記憶手段、20: 照会メッセージ発生手段、30: 回答メッセージ発生手段、AM: 回答メッセージ、CB: 共通バス、CH: 版数一致のチェックプロセス、CM: 確認メッセージ、FD: 版数記憶手段例としてのフロッピディスク装置、IB: 個別バス、IH: 照会メッセージ、I: スレーブプロセッサ番号、J: サブスレーブプロセッサ番号、M: ROMおよびRAMを含む記憶手段、M1: ROM中のマスタプロセッサの専用エリア、NP: 第1のプロセッサとしてのマスタプロセッサ、m: スレーブプロセッサの数、n: サブスレーブプロセッサの数(スレーブプロセッサ当たり)、SPI(1-1~m): 第2のプロセッサとしてのスレーブプロセッサ、SSPJ(J-1~n): 第2のプロセッサとしてのサブスレーブプロセッサ、S1~S26: 動作ステップ、VN: 版数またはバージョン数、である。

敏捷にかつ正確にすませる上にも有用である。

また、実施例の説明からも瞭解されるように本発明方式を実施するにハードウェア類をとくに追加する必要は全くなく、ソフトウェア面でもそれぞれのプロセッサの専用ROMに入れるだけの比較的簡単なものでよいので、ハードウェア、ソフトウェア両面でのいわば最低の投資で上記の顕著な効果を得ることができる。

4. 図面の簡単な説明

図はすべて本発明に関するもので、内第1図は本発明によるプログラム版数管理方式のハードウェア面の全体構成図、第2図は第1のプロセッサとしてのマスタプロセッサと第2のプロセッサとしてのスレーブプロセッサとの間のメッセージ交換の模様を例示する説明図、第3図はスレーブプロセッサとサブスレーブプロセッサとの間のメッセージ交換の模様を例示する説明図、第4図は照会メッセージおよび回答メッセージの形式を例示するメッセージでの構成図、第5図は本発明によるプログラム版数管理方式の動作をソフトウェア

-24-



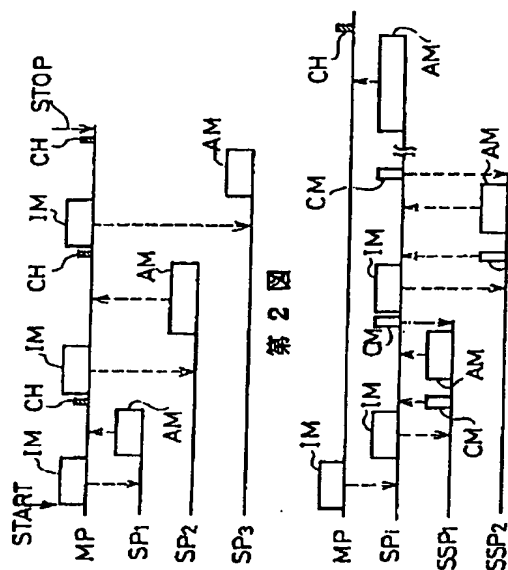


圖 3

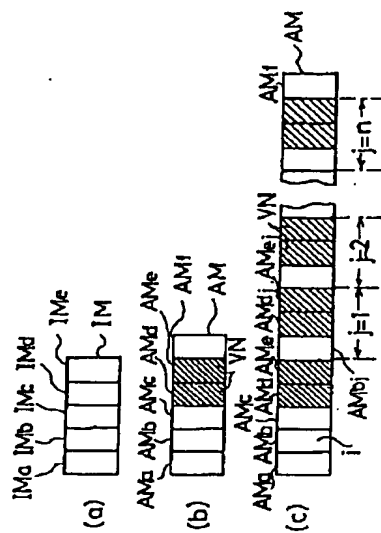
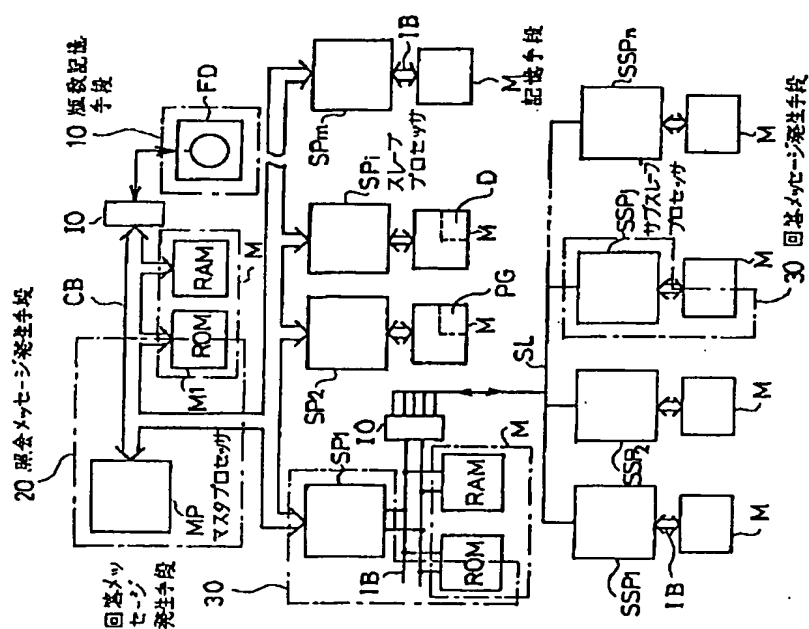
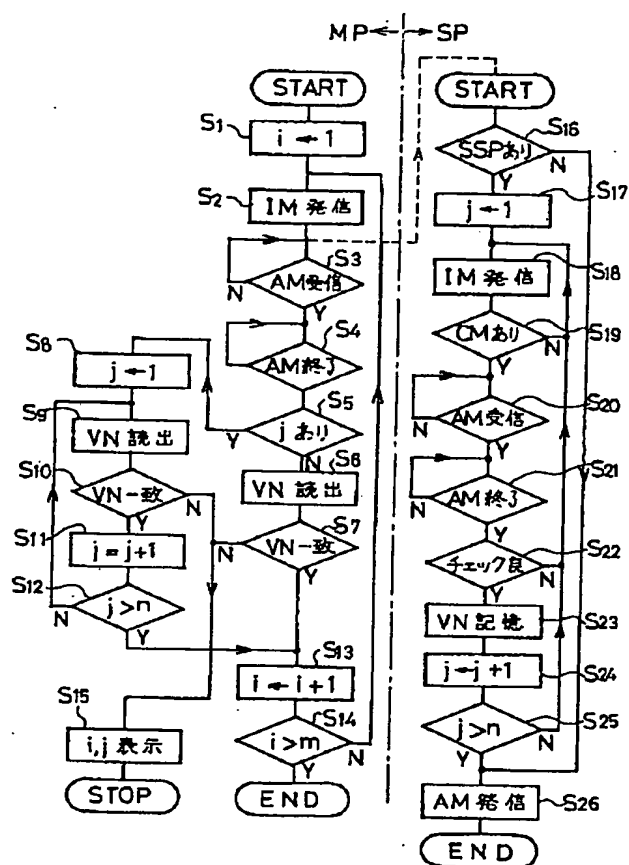


圖 4 振



圖一 搬



第 5 図